ELSEVIER
Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®
Theoretical Computer Science 306 (2003) 155–175
Theoretical Computer Science
www.elsevier.com/locate/tcs

An algebraic characterization of deterministic regular languages over infinite alphabets

Nissim Francez, Michael Kaminski*

Department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel

Received 31 October 2002; received in revised form 12 March 2003; accepted 17 March 2003
Communicated by A. Salomaa

Abstract

We state and prove an infinite alphabet counterpart of the classical Myhill–Nerode theorem.

1. Introduction

In this paper we continue the study of *finite-memory automata* introduced in [9,10] (and then addressed in [14,11]). These automata are a generalization of the classical Rabin–Scott finite-state automata [13]) to *infinite* alphabets. They were designed to recognize the "natural analog" of ordinary regular languages. While this study started as purely theoretical, since the appearance of [9,10] those ideas seem to have found their way to more practically oriented research. The key idea for the applicability is finding practical interpretations to the infinite alphabet and to the languages over it.

- In [11], members of (the infinite) alphabet Σ are interpreted as records of *communication actions*, "send" and "receive" of messages during inter-process-communica tion, words in a language L over this alphabet are MSCs, message sequence charts, capturing behaviors of the communication network.
- In [3], members of Σ are interpreted as URLs' addresses of internet sites. A word in L is interpreted as a "navigation path" in the internet, the result of some finite sequence of clicks.
- In [4] there is another internet-oriented interpretation of Σ, namely XML mark-ups of pages in a site.

* Corresponding author.
E-mail address: kaminski@cs.technion.ac.il (M. Kaminski).

0304-3975/03/$ - see front matter
doi:10.1016/S0304-3975(03)00246-9

We expect more such useful interpretations to emerge in due time, justifying further development of the theory of languages over infinite alphabet and their recognizers.

In addition to a finite set of "proper" states, finite-memory automata are equipped with a finite set of registers which in any stage of a computation (automaton's run) are either empty or contain a symbol from the infinite alphabet. By restricting the power of the automaton to copying a symbol to a register, comparing the contents of a register with an input symbol, and possibly resetting a register to empty only, without the ability to perform *any* operations, the automaton is only able to "remember" a finite set of input symbols. Thus, the languages accepted by finite-memory automata posses many of the desirable properties of regular languages.

An important facet of finite-memory automata is a certain *indistinguishability* view of the infinite alphabet embedded in the modus operandi of the automaton. The language accepted by an automaton is invariant under automorphisms of the infinite alphabet. Thus, the actual symbols occurring in the input are of no real significance. Only the initial and repetition patterns matter. This follows from the nature of the restriction to copying and comparison (reminiscent of *term-unification*). If a new symbol (i.e., one not in any register) is copied and later successfully compared, any other new symbol, having appeared in the same position, would cause the *same* transitions.

The main result of this paper is an extension of the classical Myhill–Nerode theorem to infinite alphabets. Unlike in the case of a finite alphabet, constructing a finite-memory automaton from the equivalence relation induced by the language is rather involved. The reason is that the equivalence relation induced by a language accepted by a deterministic finite-memory automaton does not have to be of a finite index. On the other hand, even though we call the model of computation "finite-memory automata," the set of its *actual* states (and, consequently, the index of the equivalence relation induced by it) is infinite, because an actual state is determined by a proper state of the automaton together with the content of its registers. Thus, when constructing an automaton from the equivalence relation induced by a language, we must assume that the language is invariant under isomorphism of the infinite alphabet to extract the state and the register components from an equivalence class.

The rest of the paper is organized as follows. In the next section we recall the definition of finite-memory automata. In Section 3 we prove some invariance properties of languages accepted by deterministic finite-memory automata and state the main result, whose proof is presented in Sections 4–8. In particular, in Section 7 we introduce an equivalent model of computation that is used for the proof of the infinite alphabet version of the Myhill–Nerode theorem. Finally, Section 9 contains some suggestions for further research.

2. Finite-memory automata

In this section we recall the definition of finite-memory automata. Let Σ be an infinite alphabet not containing # that is reserved to denote an empty register. For a word $\boldsymbol{w} = w_1 w_2 \dots w_r$ over $\Sigma \cup \{\#\}$, we define the *contents* of $\boldsymbol{w}$, denoted $[\boldsymbol{w}]$, by $[\boldsymbol{w}] = \{w_i \neq \#\colon\ i = 1, 2, \dots, r\}$. That is, $[\boldsymbol{w}]$ consists of all symbols of Σ which appear in $\boldsymbol{w}$. A word $\boldsymbol{w} = w_1w_2\ldots w_r \in (\Sigma \cup \{\#\})^*$ is called an *assignment* if $w_i = w_j$ and $i \neq j$ imply $w_i = \#\,(= w_j)$. That is, an assignment is a word in which each symbol from Σ appears at most once.

We also need the following notation. For a word $\boldsymbol{w} = w_1w_2\ldots w_n \in (\Sigma \cup \{\#\})^*$ we denote by $\#(\boldsymbol{w})$ the set of indices i for which $w_i = \#$. That is, $\#(\boldsymbol{w}) = \{i\colon\ w_i = \#\}$.

Definition 1. A *finite-memory automaton* (over Σ) or, shortly, FMA, is a tuple $\boldsymbol{A} = \langle Q, q_0, \boldsymbol{w}_0, \rho, \mu, \mathscr{F} \rangle$, where

- Q, $q_0 \in Q$, and $\mathscr{F} \subseteq Q$ are a finite set of states, the initial state, and the set of accepting states, respectively.
- $\boldsymbol{w}_0 = w_{0,1}w_{0,2}\ldots w_{0,r} \in (\Sigma \cup \{\#\})^r$, $r \geqslant 1$, is the *initial* assignment—register initialization: the symbol in the ith register is $w_{0,i}$. Recall that # denotes an empty register. That is, if $w_{0,i}$ is #, then the ith register of $\boldsymbol{A}$ is empty.
- $\rho : Q \to \{1, 2, \ldots, r\}$ is a partial function from Q to $\{1, 2, \ldots, r\}$ called the *reassignment*. The intuitive meaning of ρ is as follows. If $\boldsymbol{A}$ is in state q, $\rho(q)$ is defined, and the input symbol appears in no register, then $\boldsymbol{A}$ “forgets” the contents of the $\rho(q)$th register and copies the input symbol into that register.
- $\mu \subseteq Q \times \{1, 2, \ldots, r\} \times Q$ is the transition relation whose elements are called transitions. The intuitive meaning of μ is as follows. If the automaton is in state q, the input symbol is equal to the contents of the ith register, and $(q, i, q') \in \mu$, then $\boldsymbol{A}$ may enter state q'. In addition, if the input symbol appears in no register and is placed into the ith register ($i = \rho(q)$), then in order to enter state q' the transition relation must contain (q, i, q'). That is, the reassignment is made prior to a transition.

The initial assignment of automaton $\boldsymbol{A}$ and its length are denoted by $\boldsymbol{w}_{\boldsymbol{A}}$ and $r_{\boldsymbol{A}}$, respectively. That is, $\boldsymbol{w}_0 = \boldsymbol{w}_{\boldsymbol{A}}$ and $r = r_{\boldsymbol{A}}$.

Similar to the case of finite automata, $\boldsymbol{A}$ can be represented by its initial assignment and a finite directed graph whose nodes are states. There is an edge from q to q', if for some $i = 1, 2, \ldots, r$, $(q, i, q') \in \mu$. Such edge is labeled i. Also, if for a node q the value of ρ is defined, then q is labeled $\rho(q)$ and if $q \in \mathscr{F}$, it is labeled as such. For graph representation of finite-memory automata, see Example 1 below.

An actual state of $\boldsymbol{A}$ is a state of Q together with the contents of all registers. That is, $\boldsymbol{A}$ has infinitely many states which are pairs $(q, \boldsymbol{w})$, where $q \in Q$ and $\boldsymbol{w}$ is a word of length r—the content of the registers of $\boldsymbol{A}$. These are called configurations of $\boldsymbol{A}$. The set of all configurations of $\boldsymbol{A}$ is denoted Q^{c}. The pair $(q_0, \boldsymbol{w}_0)$, denoted q_0^{c}, is called the *initial* configuration, and the configurations with the first component in $\mathscr{F}$ are called *accepting* configurations. The set of accepting configurations is denoted $\mathscr{F}^{\mathrm{c}}$.

The transition relation μ induces the following relation μ^{c} on $Q^{\mathrm{c}} \times \Sigma \times Q^{\mathrm{c}}$.

Let $q, q' \in Q$, $\boldsymbol{w} = w_1w_2\ldots w_r$ and $\boldsymbol{w}' = w'_1w'_2\ldots w'_r$. Then the triple $((q, \boldsymbol{w}), \sigma, (q', \boldsymbol{w}'))$ belongs to μ^{c} if and only if the following conditions are satisfied.

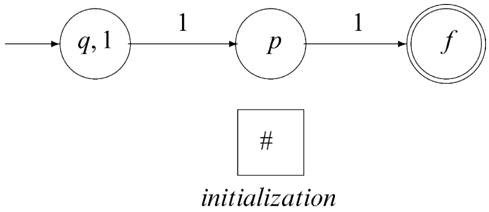
- If $\sigma = w_i \in [\boldsymbol{w}]$, then $\boldsymbol{w}' = \boldsymbol{w}$ and $(q, i, q') \in \mu$.
- If $\sigma \notin [\boldsymbol{w}]$, then $\rho(q)$ is defined, $w'_{\rho(q)} = \sigma$, $w_i = w'_i$ for each $i \neq \rho(q)$, and $(q, \rho(q), q') \in \mu$.

Let $\boldsymbol{\sigma} = \sigma_1\sigma_2\ldots\sigma_n$ be a word over Σ. A *run* of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$ consists of a sequence of configurations $q_0^{\mathrm{c}}, q_1^{\mathrm{c}}, \ldots, q_n^{\mathrm{c}}$ such that $(q_{i-1}^{\mathrm{c}}, \sigma_i, q_i^{\mathrm{c}}) \in \mu^{\mathrm{c}}$, $i = 1, 2, \ldots, n$.[1]

[1] Recall that q_0^{c} denotes the initial configuration of $\boldsymbol{A}$.

We say that $\boldsymbol{A}$ *accepts* $\boldsymbol{\sigma}$, if there exists a run $q_0^c, q_1^c, \ldots, q_n^c$ of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$ such that $q_n^c \in \mathscr{F}^c$. The set of all words accepted by $\boldsymbol{A}$ is denoted by $L(\boldsymbol{A})$ and is referred to as a *quasi-regular* language.

Example 1. Consider a one-register FMA $\boldsymbol{A}=\langle\{q,p,f\},q,\{f\},\#,\emptyset,\mu\rangle$, where $\rho(q)=1$ and $\mu=\{(q,1,p),(p,1,f)\}$. Alternatively, $\boldsymbol{A}$ can be described by the following diagram.

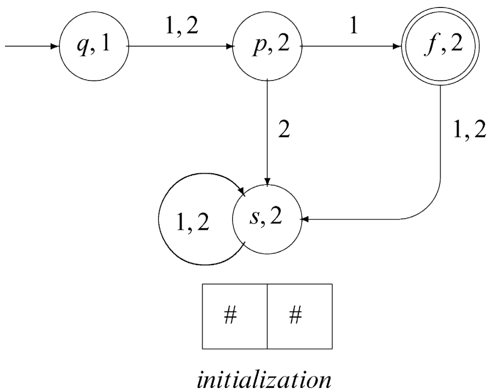


It can be easily seen that $L(\boldsymbol{A})=\{\sigma\sigma\colon\ \sigma\in\Sigma\}$: an accepting run of $\boldsymbol{A}$ on the word $\sigma\sigma$ is $(q,\#),(p,\sigma),(f,\sigma)$.

Example 2. Let $\boldsymbol{A}=\langle Q,q_0,\boldsymbol{w}_0,\rho,\mu,\mathscr{F}\rangle$ be an FMA such that # does not appear in $\boldsymbol{w}_0$ and ρ is nowhere defined. Then $L(\boldsymbol{A})$ is a regular language over $[\boldsymbol{w}_0]$. In general, since the restriction of a set of configurations to a finite alphabet is finite, the restrictions of quasi-regular languages to finite alphabets are regular, see [10, Proposition 1].

Definition 2. An FMA $\boldsymbol{A}=\langle Q,q_0,\boldsymbol{w}_0,\rho,\mu,\mathscr{F}\rangle$ is called *deterministic* or, shortly, DFMA, if ρ is everywhere defined (i.e., is a total function) and for each $q\in Q$ and each $i=1,2,\ldots,r_{\boldsymbol{A}}$ there exists exactly one $q'\in Q$ such that $(q,i,q')\in\mu$. That is, ρ is a function from Q into $\{1,2,\ldots,r_{\boldsymbol{A}}\}$ and μ can be thought of as a function from $Q\times\{1,2,\ldots,r_{\boldsymbol{A}}\}$ into Q.

Example 3. Note that FMA $\boldsymbol{A}$ in Example 1 is not deterministic, because neither is reassignment ρ defined for p and f, nor is transition function μ defined on $(q,2),(f,1)$, and $(f,2)$. Nevertheless, it can be easily "completed" to a deterministic one by introducing an additional *dead* state s and a one more register, as shown in the following diagram.



The deterministic quasi-regular languages are closed under Boolean operations, but are not closed under neither of reversing, concatenation, and Kleene star, see [10, Section 4].

3. DFMA via a congruence relation

The main result of this paper is an extension of the classical Myhill–Nerode theorem to infinite alphabets. First, let us recall the relations $\equiv_{\boldsymbol{A}}$ and $\equiv_L$ taking part in the original Myhill–Nerode Theorem.

Let $\boldsymbol{A} = \langle Q, q_0, \boldsymbol{w}_0, \rho, \mu, \mathscr{F} \rangle$ be a DFMA. We extend μ^{c} from $Q^{\mathrm{c}} \times \Sigma$ to a function $Q^{\mathrm{c}} \times \Sigma^* \to Q^{\mathrm{c}}$, also denoted μ^{c}, by putting $\mu^{\mathrm{c}}(q^{\mathrm{c}}, \varepsilon) = q^{\mathrm{c}}$ and $\mu^{\mathrm{c}}(q^{\mathrm{c}}, \boldsymbol{\sigma}\sigma) = \mu^{\mathrm{c}}(\mu^{\mathrm{c}}(q^{\mathrm{c}}, \boldsymbol{\sigma}), \sigma)$, and define a relation $\equiv_{\boldsymbol{A}}$ on Σ^* by $\boldsymbol{\sigma}' \equiv_{\boldsymbol{A}} \boldsymbol{\sigma}''$ if and only if $\mu^{\mathrm{c}}(q_0^{\mathrm{c}}, \boldsymbol{\sigma}') = \mu^{\mathrm{c}}(q_0^{\mathrm{c}}, \boldsymbol{\sigma}'')$.

Let $L \subseteq \Sigma^*$. An equivalence relation $\equiv_L$ on Σ^* is defined by $\boldsymbol{\sigma}' \equiv_L \boldsymbol{\sigma}''$ if and only if for each $\boldsymbol{\sigma} \in \Sigma^*$ the following holds: $\boldsymbol{\sigma}'\boldsymbol{\sigma} \in L$ if and only if $\boldsymbol{\sigma}''\boldsymbol{\sigma} \in L$.

Note that both $\equiv_{\boldsymbol{A}}$ are $\equiv_L$ are right congruences.

As we mentioned in the introduction, unlike in the case of a finite alphabet, constructing a finite-memory automaton from $\equiv_L$ is rather involved. This is because $\equiv_L$ does not have to be of a finite index, see Example 4. Furthermore, $\equiv_{\boldsymbol{A}}$ is not, necessarily, of a finite index either, as the set of an automaton's configurations is infinite, see Example 5. Thus, when constructing an automaton from $\equiv_L$ we must assume that the language is invariant under isomorphism of the infinite alphabet to separate an equivalence class into the state and the register components.

Example 4. Let $L = \{\sigma\sigma\colon\ \sigma \in \Sigma\}$, i.e., L is the language from Examples 1 and 3. Then for $\sigma', \sigma'' \in \Sigma$ such that $\sigma' \neq \sigma''$, $\sigma' \not\equiv_L \sigma''$, because $\sigma'\sigma'$ belongs to L, whereas $\sigma'\sigma''$ does not. Therefore, $\equiv_L$ has infinitely many equivalence classes (which are $\{\varepsilon\}$, $\{\sigma\}$ for each $\sigma \in \Sigma$, L itself, and $\Sigma^* \backslash (L \cup \Sigma \cup \{\varepsilon\})$).

Example 5. Let $\boldsymbol{A}$ be the DFMA from Example 3. Then, like in the previous example, for $\sigma', \sigma'' \in \Sigma$ such that $\sigma' \neq \sigma''$, $\sigma' \not\equiv_{\boldsymbol{A}} \sigma''$. Therefore, $\equiv_{\boldsymbol{A}}$ has infinitely many equivalence classes. In fact, $\equiv_{\boldsymbol{A}}$ coincides with $\equiv_L$.

To state the infinite alphabet counterpart of the Myhill–Nerode theorem, cf. [7, Theorem 3.9, p. 65], we shall need the following notation, definitions, and propositions.

As usual, we extend a function $F : X \to \Sigma$ to a function $X^* \to \Sigma^*$, also denoted F, by $F(\varepsilon) = \varepsilon$ and $F(\boldsymbol{\sigma}\sigma) = F(\boldsymbol{\sigma})F(\sigma)$.

Definition 3. Let $X \subseteq \Sigma$, $F : X \to \Sigma$ be one-to-one, and let $Y \subseteq \Sigma$. We say that F is *Y-preserving* if for all $\sigma \in Y \cap X$, $F(\sigma) = \sigma$.

Definition 4. Let $\equiv$ be an equivalence relation on Σ^*, $\Sigma' \subseteq \Sigma$, and let $\boldsymbol{\sigma} \in \Sigma^*$. We say that $\boldsymbol{\sigma}$ is *co-Σ'-insensitive* (with respect to $\equiv$), if for each Σ'-preserving function $F : [\boldsymbol{\sigma}] \to \Sigma$, $F(\boldsymbol{\sigma}) \equiv \boldsymbol{\sigma}$.

Example 6. Let $\boldsymbol{\sigma} \in \Sigma^*$ and let L be as in Example 4. Then $\boldsymbol{\sigma}$ is co-Σ'-insensitive with respect to $\equiv_L$, where

$$\Sigma' = \begin{cases} \{\sigma\} & \text{if } \boldsymbol{\sigma} = \sigma \in \Sigma, \\ \emptyset & \text{otherwise.} \end{cases}$$

Example 7. Let $\boldsymbol{A} = \langle Q, q_0, \boldsymbol{w}_0, \rho, \mu, \mathscr{F} \rangle$ be a DFMA, $\boldsymbol{\sigma} \in \Sigma^*$ and $\mu^{\mathrm{c}}(q_0^{\mathrm{c}}, \boldsymbol{\sigma}) = (q, \boldsymbol{w})$. It immediately follows from the proof of [10, Lemma 1] that $\boldsymbol{\sigma}$ is co-$([\boldsymbol{w}_0] \cup [\boldsymbol{w}])$-insensitive (with respect to $\equiv_{\boldsymbol{A}}$).

Definition 5. Let $\equiv$ be an equivalence relation on Σ^*, $\Sigma' \subseteq \Sigma$, and let $\mathscr{C} \subseteq \Sigma^*$ be an equivalence class of $\equiv$. We say that $\mathscr{C}$ is co-Σ'-*insensitive* (with respect to $\equiv$), if all elements of $\mathscr{C}$ are co-Σ'-*insensitive*

Example 8. Let L be as in Example 4 and let $\mathscr{C}$ be an equivalence class of $\equiv_L$. Then $\mathscr{C}$ is co-Σ'-insensitive, where

$$\Sigma' = \begin{cases} \{\sigma\} & \text{if } \mathscr{C} = \{\sigma\},\ \sigma \in \Sigma, \\ \emptyset & \text{otherwise.} \end{cases}$$

Example 9. Let $\boldsymbol{A}$, $\boldsymbol{\sigma}$, and $\boldsymbol{w}$ be as in Example 7. Then $[\boldsymbol{\sigma}]_{\equiv_{\boldsymbol{A}}}$ is co-$([\boldsymbol{w}_0] \cup [\boldsymbol{w}])$-insensitive. [2]

Definition 6. An equivalence relation $\equiv$ on Σ^* is called *co-finitely insensitive* if for each equivalence class $\mathscr{C}$ of $\equiv$ there is a finite subset Σ' of Σ such that $\mathscr{C}$ is co-Σ'-insensitive.

Example 10. Let $\boldsymbol{A}$ be as in Example 7. Then $\equiv_{\boldsymbol{A}}$ is co-finitely insensitive, because, by Example 9, for each $\boldsymbol{\sigma} \in \Sigma^*$, $[\boldsymbol{\sigma}]_{\equiv_{\boldsymbol{A}}}$ is co-$([\boldsymbol{w}_0] \cup [\boldsymbol{w}])$-insensitive, where $\mu^{\mathrm{c}}(q_0^{\mathrm{c}}, \boldsymbol{\sigma}) = (q, \boldsymbol{w})$.

Example 11. Let $\equiv$ be an equivalence relation on Σ^* such that for some two different equivalence classes $\mathscr{C}'$ and $\mathscr{C}''$ of $\equiv$ both sets $\mathscr{C}' \cap \Sigma$ and $\mathscr{C}'' \cap \Sigma$ are infinite. Then $\equiv$ is not co-finitely insensitive. Indeed, assume to the contrary that for some finite subset Σ' of Σ equivalence class $\mathscr{C}'$ is co-Σ'-insensitive. Let $\sigma' \in (\mathscr{C}' \cap \Sigma) \backslash \Sigma'$ and let $\sigma'' \in (\mathscr{C}'' \cap \Sigma) \backslash \Sigma'$. Such symbols σ' and σ'' exists because both $\mathscr{C}' \cap \Sigma$ and $\mathscr{C}'' \cap \Sigma$ are infinite and Σ' is finite. Let F be a Σ'-preserving function such that $F(\sigma') = \sigma''$. Then $F(\sigma') \not\equiv \sigma'$.

Definition 7. Let $\Sigma' \subseteq \Sigma$. An equivalence relation $\equiv$ on Σ^* is called *co-Σ'-invariant* if for each $\boldsymbol{\sigma}', \boldsymbol{\sigma}'' \in \Sigma^*$ such that $\boldsymbol{\sigma}' \equiv \boldsymbol{\sigma}''$ the following holds. For each Σ'-preserving function $F : [\boldsymbol{\sigma}'] \cup [\boldsymbol{\sigma}''] \to \Sigma$, $F(\boldsymbol{\sigma}') \equiv F(\boldsymbol{\sigma}'')$.

Example 12. Let L be as in Example 4. Then $\equiv_L$ is co-$\emptyset$-invariant.

[2] As usual, $[\boldsymbol{\sigma}]_{\equiv_{\boldsymbol{A}}}$ denotes the equivalence class of $\boldsymbol{\sigma}$ with respect to $\equiv_{\boldsymbol{A}}$.

Example 13. Let $\boldsymbol{A}$ be as in Example 7. It follows from the proof of [10, Lemma 1] that $\equiv_{\boldsymbol{A}}$ is co-$[\boldsymbol{w}_0]$-invariant.

Example 14. Let $\Sigma' \subseteq \Sigma$ and let $\equiv$ be a co-Σ'-invariant equivalence relation on Σ^*. Then $[\varepsilon]_{\equiv}$ is co-Σ'-insensitive. Indeed, let $\boldsymbol{\sigma} \in [\varepsilon]_{\equiv}$ and let $F : [\boldsymbol{\sigma}] \to \Sigma$ be a Σ'-preserving function. Then

$$F(\boldsymbol{\sigma}) \equiv F(\varepsilon) = \varepsilon \equiv \boldsymbol{\sigma},$$

where the first equivalence follows from co-Σ'-invariance of $\equiv$.

Let $\Sigma' \subseteq \Sigma$ and let $\equiv$ be a co-Σ'-invariant equivalence relation on Σ^*. We denote by $\equiv^{\Sigma'}$ the derived relation on Σ^* that is defined as follows.

$\sigma' \equiv^{\Sigma'} \sigma''$ if and only if for some Σ'-preserving function F, $F(\sigma') \equiv \sigma''$.

Since compositions and inverses of Σ'-preserving functions are also Σ'-preserving, $\equiv^{\Sigma'}$ is an equivalence relation on Σ^*.

Example 15. Let $\boldsymbol{A}$ be as in Example 7. Then $\equiv_{\boldsymbol{A}}^{[\boldsymbol{w}_0]}$ is of a finite index. Indeed, let $\boldsymbol{\sigma}', \boldsymbol{\sigma}'' \in \Sigma^*$ and let $\mu^{\mathrm{c}}(q_0, \boldsymbol{\sigma}') = (q', \boldsymbol{w}')$, $\boldsymbol{w}' = w_1' w_2' \ldots w_r'$, and $\mu^{\mathrm{c}}(q_0, \boldsymbol{\sigma}'') = (q'', \boldsymbol{w}'')$, $\boldsymbol{w}'' = w_1'' w_2'' \ldots w_r''$. Then $\boldsymbol{\sigma}' \not\equiv_{\boldsymbol{A}}^{[\boldsymbol{w}_0]} \boldsymbol{\sigma}''$ if and only if $q' \neq q''$ or for some $i = 1, 2, \ldots, r$ either of w_i' or w_i'' is in $[\boldsymbol{w}_0] \cup \{\#\}$, but $w_i' \neq w_i''$. Thus, there can be only finitely many pairwise non-equivalent words.

Theorem 1 below is the infinite alphabet counterpart of the Myhill–Nerode theorem, cf. [7, Theorem 3.9, p. 65]. To state Theorem 1 we need the following definition.

Definition 8. Let $\Sigma' \subseteq \Sigma$. A language L over Σ is called *co-Σ'-invariant* if for each $\boldsymbol{\sigma} \in L$ and for each Σ'-preserving function $F : [\boldsymbol{\sigma}] \to \Sigma$, $F(\boldsymbol{\sigma}) \in L$.

Example 16. Let L be as in Example 4. Then L is co-$\emptyset$-invariant.

Example 17. Let $\boldsymbol{A}$ be as in Example 7. Then, by Kaminski and Francez [10, Proposition 2], $L(\boldsymbol{A})$ is co-$[\boldsymbol{w}_0]$-invariant.

Theorem 1. *For a language L over Σ the following three conditions are equivalent.*

I *L is a quasi-regular language.*

II *There exist a finite subset Σ' of Σ and a co-finitely insensitive and co-Σ'-invariant right congruence $\equiv$ on Σ^* such that $\equiv^{\Sigma'}$ is of a finite index and L is a union of equivalence classes of $\equiv^{\Sigma'}$.*

III *$\equiv_L$ is co-finitely insensitive, L is co-Σ'-invariant for some finite subset Σ' of Σ, and $\equiv_L^{\Sigma'}$ is of a finite index.*

We shall prove that I implies III, III implies II, and II implies I. The proofs are presented in Sections 4, 5, and 8, respectively. Whereas the proofs of implications I $\Rightarrow$ III and III $\Rightarrow$ II are easy generalizations of the corresponding proofs for a finite alphabet, the proof of II $\Rightarrow$ I is rather involved. In particular, it is based on an alternative (but equivalent) model of computation called *Reset Finite-Memory Automata* introduced in Section 8. When constructing a deterministic reset finite-memory automaton from L we shall use both co-finite insensitivity of $\equiv_L$ and co-Σ'-invariance of L to "separate" an equivalence class of $\equiv_L$ into the state and the register components. Namely, equivalence classes of $\equiv_L^{\Sigma'}$ will serve as the automaton states, and co-finite insensitivity of $\equiv_L$ (together with a finite index of $\equiv_L^{\Sigma'}$) will provide an upper bound on the number of the automaton registers, see Proposition 3 in Section 7.

4. Proof of I ⇒ III

Let $L = L(\boldsymbol{A})$ for a DFMA $\boldsymbol{A} = \langle Q, q_0, \boldsymbol{w}_0, \rho, \mu, \mathscr{F} \rangle$. Then, by Example 17, L is co-$[\boldsymbol{w}_0]$-invariant. To show that $\equiv_L$ is co-finitely insensitive, we need Proposition 1 below.

Proposition 1. $\equiv_L$ *is co*-$[\boldsymbol{w}_0]$-*invariant.*

Proof. Let $\boldsymbol{\sigma}', \boldsymbol{\sigma}'' \in \Sigma^*$ and let $F : [\boldsymbol{\sigma}'] \cup [\boldsymbol{\sigma}''] \to \Sigma$ be a $[\boldsymbol{w}_0]$-preserving function. Then, by the definition of $\equiv_L$,

$$\boldsymbol{\sigma}' \equiv_L \boldsymbol{\sigma}''$$

if and only if

$$\text{for all } \boldsymbol{\sigma} \in \Sigma^*, \boldsymbol{\sigma}'\boldsymbol{\sigma} \in L \text{ if and only if } \boldsymbol{\sigma}''\boldsymbol{\sigma} \in L,$$

which, by co-$[\boldsymbol{w}_0]$-invariance of L, holds if and only if

$$\text{for all } \boldsymbol{\sigma} \in \Sigma^*, F(\boldsymbol{\sigma}')F(\boldsymbol{\sigma}) \in L \text{ if and only if } F(\boldsymbol{\sigma}'')F(\boldsymbol{\sigma}) \in L$$

which, by substitution of $F^{-1}(\boldsymbol{\sigma})$ for $\boldsymbol{\sigma}$, is equivalent to

$$\text{for all } \boldsymbol{\sigma} \in \Sigma^*, F(\boldsymbol{\sigma}')\boldsymbol{\sigma} \in L \text{ if and only if } F(\boldsymbol{\sigma}'')\boldsymbol{\sigma} \in L$$

which, by the definition of $\equiv_L$, is

$$F(\boldsymbol{\sigma}') \equiv_L F(\boldsymbol{\sigma}''). \qquad \square$$

Now we can show that $\equiv_L$ is co-finitely insensitive. Namely, we shall prove that if $\mu^c(q_0, \boldsymbol{\sigma}) = (q, \boldsymbol{w})$, then the $\equiv_L$ equivalence class of $\boldsymbol{\sigma}$ is co-$([\boldsymbol{w}] \cup [\boldsymbol{w}_0])$-insensitive. Let $F : [\boldsymbol{\sigma}] \to \Sigma$ be a $([\boldsymbol{w}] \cup [\boldsymbol{w}_0])$-preserving function and let $\boldsymbol{\sigma}' \in [\boldsymbol{\sigma}]_{\equiv_L}$.

By Proposition 1, $\equiv_L$ is co-$[\boldsymbol{w}_0]$-invariant, implying

$$F(\boldsymbol{\sigma}') \equiv_L F(\boldsymbol{\sigma}). \tag{1}$$

Since, by Example 7, $\boldsymbol{\sigma}$ is co-$([\boldsymbol{w}_0] \cup [\boldsymbol{w}])$-insensitive, $F(\boldsymbol{\sigma}) \equiv_{\boldsymbol{A}} \boldsymbol{\sigma}$. Exactly like in the proof of [7, Theorem 3.9, p. 65] it can be shown that $\equiv_{\boldsymbol{A}}$ is a refinement of $\equiv_L$.[3]

[3] That is, $\boldsymbol{\sigma}' \equiv_{\boldsymbol{A}} \boldsymbol{\sigma}''$ implies $\boldsymbol{\sigma}' \equiv_L \boldsymbol{\sigma}''$.

Thus, $F(\boldsymbol{\sigma}) \equiv_{\boldsymbol{A}} \boldsymbol{\sigma}$ implies

$$F(\boldsymbol{\sigma}) \equiv_L \boldsymbol{\sigma}. \tag{2}$$

Also, since $\boldsymbol{\sigma}' \in [\boldsymbol{\sigma}]_{\equiv_L}$, by definition,

$$\boldsymbol{\sigma} \equiv_L \boldsymbol{\sigma}'. \tag{3}$$

Now the equivalence $F(\boldsymbol{\sigma}') \equiv_L \boldsymbol{\sigma}'$ follows from (1)–(3).

Finally, since, as we saw in the proof of I $\Rightarrow$ II, $\equiv_{\boldsymbol{A}}^{[\boldsymbol{w}_0]}$ is of a finite index, to show that $\equiv_L^{\Sigma'}$ is of a finite index, it suffices to show that $\equiv_{\boldsymbol{A}}^{[\boldsymbol{w}_0]}$ is refinement of $\equiv_L^{[\boldsymbol{w}_0]}$. So, assume $\boldsymbol{\sigma}' \equiv_{\boldsymbol{A}}^{[\boldsymbol{w}_0]} \boldsymbol{\sigma}''$. That is, for some $[\boldsymbol{w}_0]$-preserving function $F : [\boldsymbol{\sigma}'] \to \Sigma$, $F(\boldsymbol{\sigma}') \equiv_{\boldsymbol{A}} \boldsymbol{\sigma}''$. Since $\equiv_{\boldsymbol{A}}$ is refinement of $\equiv_L$, $F(\boldsymbol{\sigma}') \equiv_L \boldsymbol{\sigma}''$. Whence, $\boldsymbol{\sigma}' \equiv_L^{[\boldsymbol{w}_0]} \boldsymbol{\sigma}''$, which completes the proof.

5. Proof of III $\Rightarrow$ II

It suffices to show that $\equiv_L$ is a co-Σ'-invariant right congruence and L is the union of a number of equivalence classes of $\equiv_L^{\Sigma'}$.

The proof of right congruence of $\equiv_L$ is exactly like the corresponding proof in [7, Theorem 3.9, p. 65] that also works for infinite alphabets.

We show next that $\equiv_L$ is co-Σ'-invariant. Let

$$\boldsymbol{\sigma}' \equiv_L \boldsymbol{\sigma}'' \tag{4}$$

and let $F : [\boldsymbol{\sigma}'] \cup [\boldsymbol{\sigma}''] \to \Sigma$ be a Σ'-preserving function. We have to show that for all $\boldsymbol{\sigma} \in \Sigma^*$, $F(\boldsymbol{\sigma}')\boldsymbol{\sigma} \in L$ if and only if $F(\boldsymbol{\sigma}'')\boldsymbol{\sigma} \in L$. We have

$$F(\boldsymbol{\sigma}')\boldsymbol{\sigma} \in L$$

if and only if

$\boldsymbol{\sigma}' F^{-1}(\boldsymbol{\sigma}) \in L$, where F^{-1} denotes (an extension of) the inverse of F,

if and only if

$$\boldsymbol{\sigma}'' F^{-1}(\boldsymbol{\sigma}) \in L$$

if and only if

$$F(\boldsymbol{\sigma}'')\boldsymbol{\sigma} \in L,$$

where the first and the last equivalences follow from Σ' invariance of F and F^{-1} and co-Σ' invariance of L, and the middle equivalence follows from (4).

Finally, we show that $L = \bigcup_{\boldsymbol{\sigma} \in L} [\boldsymbol{\sigma}]_{\equiv_L^{\Sigma'}}$. Obviously, $L \subseteq \bigcup_{\boldsymbol{\sigma} \in L} [\boldsymbol{\sigma}]_{\equiv_L^{\Sigma'}}$. For the proof of the converse inclusion, let $\boldsymbol{\sigma}' \in \bigcup_{\boldsymbol{\sigma} \in L} [\boldsymbol{\sigma}]_{\equiv_L^{\Sigma'}}$. That is, for some $\boldsymbol{\sigma} \in L$ and for some Σ'-preserving function $F : [\boldsymbol{\sigma}'] \to \Sigma$, $F(\boldsymbol{\sigma}') \equiv_L \boldsymbol{\sigma}$. It follows from the definition of $\equiv_L$

that $F(\boldsymbol{\sigma}') \in L$. Since, the inverse F^{-1} of F is also Σ'-preserving, $\boldsymbol{\sigma}' \in L$ follows from Σ'-invariance of L.

6. Insensitive and invariant relations

This section contains some properties of insensitive and invariant relations we shall use in the proof of implication II $\Rightarrow$ I.

Proposition 2. *Let $\equiv$ be an equivalence relation on Σ^*, Σ_j, $j \in J$, be subsets of Σ, and let $\mathscr{C}$ be an equivalence class of $\equiv$ that is co-Σ_j-insensitive for all $j \in J$. Then $\mathscr{C}$ is co-$\bigcap_{j \in J} \Sigma_j$-insensitive.*

Proof. Let $\boldsymbol{\sigma} \in \mathscr{C}$, $[\boldsymbol{\sigma}] \setminus \bigcap_{j \in J} \Sigma_j = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ and let $\{\delta_1, \delta_2, \ldots, \delta_n\}$ be any n-element subset of Σ disjoint from $\{\sigma_1, \sigma_2, \ldots, \sigma_n\} \cup \{(F(\sigma_1), F(\sigma_2), \ldots, F(\sigma_n)\}$. (Such a set $\{\delta_1, \delta_2, \ldots, \delta_n\}$ exists, because Σ is infinite.) Let $G : [\boldsymbol{\sigma}] \to \Sigma$ be defined by

$$G(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \bigcap\limits_{j \in J} \Sigma_j, \\ \delta_i & \text{if } \sigma = \sigma_i, \ i = 1, 2, \ldots, n \end{cases}$$

and let $H : [G(\boldsymbol{\sigma})] \to \Sigma$ be defined by

$$H(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \bigcap\limits_{j \in J} \Sigma_j, \\ F(\sigma_i) & \text{if } \sigma = \delta_i, \ i = 1, 2, \ldots, n. \end{cases}$$

By definition, both G and H are $\bigcap_{j \in J} \Sigma_j$-preserving and $F = H \circ G$. Thus, it suffices to show that $G(\boldsymbol{\sigma}) \equiv \boldsymbol{\sigma}$ and $H(G(\boldsymbol{\sigma})) \equiv G(\boldsymbol{\sigma})$.

For the proof of $G(\boldsymbol{\sigma}) \equiv \boldsymbol{\sigma}$ we proceed as follows. Let $g_i : [\boldsymbol{\sigma}] \to \Sigma$ be defined by

$$g_i(\sigma) = \begin{cases} \delta_i & \text{if } \sigma = \sigma_i, \\ \sigma & \text{otherwise.} \end{cases}$$

Then g_i is $\bigcap_{j \in J} \Sigma_j$-preserving. Let $G_i = g_i \circ g_{i-1} \circ \cdots \circ g_1$. Then $G = G_n$ and it suffices to show that $g_i(G_{i-1}(\boldsymbol{\sigma})) \equiv G_{i-1}(\boldsymbol{\sigma})$, $i = 1, 2, \ldots, n$. The proof is by induction on i.

Basis: i=0. Since G_0 is the identity function, the claim follows from reflexivity of $\equiv$.

Induction step: Let $i > 1$ and assume that for all $k < i$, $g_k(G_{k-1}(\boldsymbol{\sigma}) \equiv G_{k-1}(\boldsymbol{\sigma})$. Whence, by transitivity of $\equiv$, $G_{i-1}(\boldsymbol{\sigma}) \in \mathscr{C}$. By the definition of the set $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, for some $j \in J$, $\sigma_i \notin \Sigma_j$. Then $g_i(G_{i-1}(\boldsymbol{\sigma})) \equiv G_{i-1}(\boldsymbol{\sigma})$, because, in particular, g_i is Σ_j-preserving.

The proof of $H(G(\boldsymbol{\sigma})) \equiv G(\boldsymbol{\sigma})$ is similar to the above. Let $h_i : [\boldsymbol{\sigma}] \to \Sigma$ be defined by

$$h_i(\sigma) = \begin{cases} F(\delta_i) & \text{if } \sigma = \delta_i, \\ \sigma & \text{otherwise} \end{cases}$$

Then h_i is $\bigcap_{j \in J} \Sigma_j$-preserving. Let $H_i = h_i \circ h_{i-1} \circ \cdots \circ h_1$. Then $H = H_n$ and it suffices to show that $h_i(H_{i-1}(G(\boldsymbol{\sigma})) \equiv H_{i-1}(G(\boldsymbol{\sigma}))$, $i = 1, 2, \ldots, n$. Again, the proof is by induction on i.

Basis: $i = 0$. Since H_0 is the identity function, the claim follows from reflexivity of $\equiv$.

Induction step: Let $i > 1$ and assume that for all $k < i$, $h_k(H_{k-1}(G(\boldsymbol{\sigma})) \equiv H_{k-1}(G(\boldsymbol{\sigma}))$. Since, as we showed above, $G(\boldsymbol{\sigma}) \in \mathscr{C}$, by transitivity of $\equiv$, $H_{i-1}(G(\boldsymbol{\sigma})) \in \mathscr{C}$ as well. By the definition of the set $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, for some $j \in J$, $F(\sigma_i) \notin \Sigma_j$. Therefore, $h_i(H_{i-1}(G(\boldsymbol{\sigma}))) \equiv H_{i-1}(G(\boldsymbol{\sigma}))$, because, in particular, h_i is Σ_j-preserving. □

Corollary. *Let $\equiv$ be an equivalence relation on Σ^* and let $\mathscr{C}$ be an equivalence class of $\equiv$. Then the set*

$$\{\Sigma' \subseteq \Sigma\colon\ \mathscr{C} \text{ is co-}\Sigma'\text{-insensitive}\} \tag{5}$$

has a least element (with respect to inclusion).

Proof. Since, by definition, $\mathscr{C}$ is co-Σ-insensitive, the least element of (5) is the intersection of all its elements that, by Proposition 2, belongs to (5). □

We denote the least element of (5) by $\Sigma^{\mathscr{C}}$.

Example 18. Let L be as in Example 4 and let $\mathscr{C}$ be an equivalence class of $\equiv_L$. Then $\Sigma^{\mathscr{C}} = \Sigma'$, where

$$\Sigma' = \begin{cases} \{\sigma\} & \text{if } \mathscr{C} = \{\sigma\}, \sigma \in \Sigma, \\ \emptyset & \text{otherwise.} \end{cases}$$

Definition 9. Let r be a non-negative integer. An equivalence relation $\equiv$ on Σ^* is called *co-r-insensitive* if for each equivalence class $\mathscr{C}$ of $\equiv$ there is a finite subset Σ' of Σ that contains at most r elements such that $\mathscr{C}$ is co-Σ'-insensitive.

Example 19. Let L be as in Example 4. Then, by Example 18, $\equiv_L$ is co-1-insensitive.

Example 20. Let $\boldsymbol{A}$, $\boldsymbol{\sigma}$, and $\boldsymbol{w}$ be as in Example 7 and let r be the number of elements of $[\boldsymbol{w}_0] \cup [\boldsymbol{w}]$. Then $[\boldsymbol{\sigma}]_{\equiv_{\boldsymbol{A}}}$ is co-r-insensitive.

Remark 1. Let r be a non-negative integer. Obviously, each co-r-insensitive relation is also co-finitely insensitive.

Proposition 3. *Let $\Sigma' \subseteq \Sigma$ and let $\equiv$ be a co-finitely insensitive and co-Σ'-invariant equivalence relation on Σ^* such that $\equiv^{\Sigma'}$ is of a finite index. Then $\equiv$ is co-r-insensitive for some positive integer r.*

Proof. We observe first that if $\boldsymbol{\sigma} \in \Sigma^*$ is co-X-insensitive and F is a Σ'-preserving function, then $F(\boldsymbol{\sigma})$ is co-$F(X)$-insensitive. Indeed, let G be an $F(X)$-preserving function. Then the composition $F^{-1} \circ G \circ F$ of F,G, and F^{-1} is X-preserving, implying $F^{-1}(G(F(\boldsymbol{\sigma}))) \equiv \boldsymbol{\sigma}$. Since F is Σ'-preserving, $G(F(\boldsymbol{\sigma})) \equiv F(\boldsymbol{\sigma})$.

Now, let $\boldsymbol{\sigma}', \boldsymbol{\sigma}'', \ldots, \boldsymbol{\sigma}_n$ be representatives of the equivalence classes of $\equiv^{\Sigma'}$. Let $[\boldsymbol{\sigma}_i]_\equiv$ be co-X_i-insensitive for some finite subset X_i of Σ, $i = 1, 2, \ldots, n$. Let $\boldsymbol{\sigma} \in \Sigma^*$ and let $\boldsymbol{\sigma}' \in [\boldsymbol{\sigma}]_\equiv$. There exists an $i = 1, 2, \ldots, n$ and a Σ'-preserving function F such that $F(\boldsymbol{\sigma}) \in [\boldsymbol{\sigma}_i]_\equiv$, which, in turn, implies $F(\boldsymbol{\sigma}') \in [\boldsymbol{\sigma}_i]_\equiv$. By the observation above, $F(\boldsymbol{\sigma}')$ is co-$F^{-1}(X_i)$-insensitive. Thus, $[\boldsymbol{\sigma}]_\equiv$ is co-$F^{-1}(X_i)$-insensitive as well and we can put r be the maximum of the cardinalities of the X_is. $\square$

Proposition 4. *Let $\Sigma' \subseteq \Sigma$ and let $\equiv$ be a co-Σ'-invariant right congruence on Σ^*. Let $\boldsymbol{\sigma} \in \Sigma^*$, and $\sigma \in \Sigma$. Then $\Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv} \subseteq \Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \{\sigma\} \cup \Sigma'$.*

Proof. We shall prove first that $\boldsymbol{\sigma}\sigma$ is co-$(\Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \{\sigma\})$-insensitive. Let $F : [\boldsymbol{\sigma}\sigma] \to \Sigma$ be a $(\Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \{\sigma\})$-preserving function. Then, in particular, $F(\sigma) = \sigma$, implying $F(\boldsymbol{\sigma}\sigma) = F(\boldsymbol{\sigma})\sigma$. Since F is $(\Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \{\sigma\})$-preserving, $F(\boldsymbol{\sigma}) \equiv \boldsymbol{\sigma}$ and the result follows from right congruence of $\equiv$.

Now let $\boldsymbol{\sigma}' \in [\boldsymbol{\sigma}\sigma]_\equiv$ and let $F : [\boldsymbol{\sigma}\sigma] \to \Sigma$ be a $(\Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \{\sigma\} \cup \Sigma')$-preserving function. Then

$$F(\boldsymbol{\sigma}') \equiv F(\boldsymbol{\sigma}\sigma) \equiv \boldsymbol{\sigma}\sigma,$$

where the first equivalence follows from co-Σ'-invariance of $\equiv$ and the second equivalence follows from co-$(\Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \{\sigma\})$-insensitivity of $\boldsymbol{\sigma}\sigma$ observed in the beginning of the proof. $\square$

7. Reset finite-memory automata

In this section we introduce a model of computation (equivalent to FMA) called a *reset finite-memory automaton* or, shortly, RFMA, that is more convenient for the proof of implication II $\Rightarrow$ I. The main feature of RFMAs is that they allow to reset some registers after reading the input symbol. This feature will be used for computation of $\Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv}$ from $\Sigma^{[\boldsymbol{\sigma}]_\equiv}$.

Definition 10. A *reset finite-memory automaton* (over Σ) is a tuple $\boldsymbol{A}=\langle Q,q_0,\boldsymbol{w}_0,\mu,\mathscr{F}\rangle$, where

- Q, $q_0 \in Q$, and $\mathscr{F} \subseteq Q$ are a finite set of states, the initial state, and the set of accepting states, respectively.
- $\boldsymbol{w}_0 = w_{0,1}w_{0,2}\ldots w_{0,r} \in (\Sigma\cup\{\#\})^r$, $r \geqslant 1$, is the *initial* assignment such that for at least one $i\in\{1,2,\ldots,r\}$, $w_{0,i}=\#$.
- $\mu \subseteq Q\times\{1,2,\ldots,r\}\times(2^{\{1,2,\ldots,r\}}\backslash\{\emptyset\})\times Q$ is the transition relation whose elements are called transitions. The intuitive meaning of μ is as follows. If the automaton is in state q reading symbol σ and there is a transition $(q,i,I,q')\in\mu$ such that register i contains σ, then the automaton can enter state q' and empty (reset) the registers whose indices belong to I.

Like in the case of FMA, an actual state of $\boldsymbol{A}$ is a configuration. The transition relation μ induces the transition relation μ^{c} on $Q^{\mathrm{c}}\times\Sigma\times Q^{\mathrm{c}}$.

Let $q,q'\in Q$, $\boldsymbol{w}=w_1w_2\ldots w_r$ and $\boldsymbol{w}'=w'_1w'_2\ldots w'_r$. Then the triple $((q,\boldsymbol{w}),\sigma,(q',\boldsymbol{w}'))$ belongs to μ^{c} if and only if there is a transition $(q,i,I,q')\in\mu$ such that the following conditions are satisfied:

- If $\sigma=w_i\in[\boldsymbol{w}]$, then

$$w'_k=\begin{cases} w_k & \text{if } k\notin I,\\ \# & \text{otherwise.}\end{cases}$$

- If $\sigma\notin[\boldsymbol{w}]$, then
  - $i=\min(\#(\boldsymbol{w}))$,
  - $w'_i=\begin{cases} \sigma & \text{if } i\notin I,\\ \# & \text{otherwise}\end{cases}$

    and
  - for $k\neq i$, $w'_k=\begin{cases} w_k & \text{if } k\notin I,\\ \# & \text{otherwise.}\end{cases}$

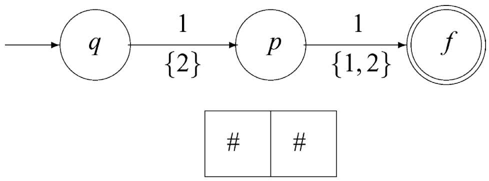
  That is, if $\sigma\notin[\boldsymbol{w}]$, then $\boldsymbol{A}$ first stores σ in the empty register with the smallest index (i), makes a move, and, finally, resets the registers whose indices are in I.

Let $\boldsymbol{\sigma}=\sigma_1\sigma_2\ldots\sigma_n$ be a word over Σ. A *run* of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$ consists of a sequence of configurations $q_0^{\mathrm{c}},q_1^{\mathrm{c}},\ldots,q_n^{\mathrm{c}}$ such that $(q_{i-1}^{\mathrm{c}},\sigma_i,q_i^{\mathrm{c}})\in\mu^{\mathrm{c}}$, $i=1,2,\ldots,n$.

We say that $\boldsymbol{A}$ *accepts* $\boldsymbol{\sigma}$, if there exists a run $q_0^{\mathrm{c}},q_1^{\mathrm{c}},\ldots,q_n^{\mathrm{c}}$ of $\boldsymbol{A}$ on $\boldsymbol{\sigma}$ such that $q_n^{\mathrm{c}}\in\mathscr{F}^{\mathrm{c}}$.

Like finite-memory automata, $\boldsymbol{A}$ can be represented by its initial assignment and a directed graph whose nodes are states. There is an edge from q to q', if for some $i=1,2,\ldots,r$ and $I\subseteq\{1,2,\ldots,r\}$, $(q,i,I,q')\in\mu$. Such edge is labeled with both i and I.

Example 21. Consider a two-register RFMA $\boldsymbol{A}=\langle\{q,p,f\},q,\#\#,\mu,\{f\}\rangle$, where $\mu=\{(q,1,\{2\},p),(p,1,\{2\},f)\}$. Alternatively, $\boldsymbol{A}$ can be described by the following diagram.



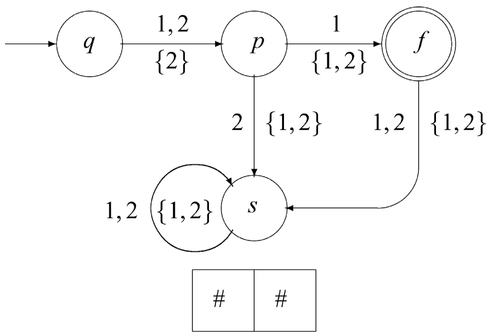
initialization

It can be easily seen that $L(A) = \{\sigma\sigma\colon\ \sigma \in \Sigma\}$: an accepting run of A on the word $\sigma\sigma$ is $(q, \#\#), (p, \sigma\#), (f, \sigma\#)$.

Definition 11. An RFMA $A = \langle Q, q_0, w_0, \mu, \mathscr{F}\rangle$ is called *deterministic* or, shortly, DRFMA, if for each $q \in Q$ and each $i = 1, 2, \ldots, r_A$ there exists exactly one non-empty $I \subseteq \{1, 2, \ldots, r_A\}$ and exactly one $q' \in Q$ such that $(q, i, I, q') \in \mu$. That is, μ can be thought of as a function from $Q \times \{1, 2, \ldots, r_A\}$ into $(2^{\{1,2,\ldots,r_A\}} \setminus \{\emptyset\}) \times Q$.

Remark 2. By definition, # appears in the initial assignment and the set of indices of the reset registers is not empty. Thus, the next move of an DRFMA is always defined.

Example 22. Note that RFMA A in Example 21 is not deterministic, because there is no move from f. Nevertheless, it can be easily "completed" to a deterministic one by introducing an additional "dead" state s and a one more register, as shown in the following diagram.



initialization

Theorem 2. *If a language is accepted by a DRFMA then it is accepted by a DFMA.*

Remark 3. Actually, it can be readily seen that (D)RFMAs and (D)FMAs accept the same languages, respectively. In particular, the proof of Theorem 2 extends to the non-deterministic case in a straightforward manner.

Proof of Theorem 2. Given an r-register DRFMA $\boldsymbol{A}=\langle Q,q_0,\boldsymbol{w}_0,\mu,\mathscr{F}\rangle$ we construct an DFMA $\hat{\boldsymbol{A}}=\langle\hat{Q},\hat{q}_0,\hat{\boldsymbol{w}}_0,\rho,\hat{\mu},\hat{\mathscr{F}}\rangle$ whose components are defined as follows.

- $\hat{Q}=Q\times(2^{\{1,2,\ldots,r\}}\setminus\{\emptyset\})\times S_r$, where S_r denotes the group of all permutations of $(1,2,\ldots,r)$. The intuitive meaning of the second component of a state of $\hat{\boldsymbol{A}}$ is that it consists of all indices of the empty registers of the corresponding configuration of $\boldsymbol{A}$ and the "permutation" component provides a correspondence between the contents of the registers of $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$.
- $\hat{q}_0=(q_0,\#(\boldsymbol{w}_0)),\mathrm{Id})$, where Id denotes the identity permutation.
- $\hat{\boldsymbol{w}}_0=\boldsymbol{w}_0$.
- $\rho((q,I,\pi))=\min(I)$.
- $\hat{\mathscr{F}}=\mathscr{F}\times(2^{\{1,2,\ldots,r\}}\setminus\{\emptyset\})\times S_r$.
- Finally, $\hat{\mu}$ is defined as follows. Let $\mu(q',i)=(I,q'')$, I' be a non-empty subset of $\{1,2,\ldots,r\}$, and let $\pi'\in S_r$. Then $\hat{\mu}((q',I',\pi'),i)=(q'',I'',\pi'')$, where

  $$I''=\begin{cases} I'\cup I & \text{if } i\notin I',\\ (I'\setminus\{\min(I')\})\cup I & \text{otherwise}\end{cases}$$

  and

  $$\pi''=\begin{cases} \pi' & \text{if } i\notin I',\\ (i,\min(I'))\pi' & \text{otherwise.}\end{cases}$$

  As usual, (i_1,i_2) denotes the transposition of i_1 and i_2, i.e., the permutation of $(1,2,\ldots,r)$ that switches between positions i_1 and i_2 and leaves all other positions intact.

To explain the intuitive meaning of $\hat{\mu}$ we shall need the following notation. Let I be a non-empty subset of $\{1,2,\ldots,r\}$ and let $\boldsymbol{w}=w_1w_2\ldots w_r\in(\Sigma\cup\{\#\})^*$. For $i=1,2,\ldots,r$ we define a symbol $w_i^I\in[\boldsymbol{w}]\cup\{\#\})$ by

$$w_i^I=\begin{cases} w_i & \text{if } i\notin I,\\ \# & \text{otherwise}\end{cases}$$

and we define a word $\boldsymbol{w}^I$ by $\boldsymbol{w}^I=w_1^Iw_2^I\ldots w_r^I$.

Then it follows from the corresponding definitions that $\hat{\mu}^{\mathrm{c}}((q',I',\pi'),\boldsymbol{w}')=((q'',I'',\pi''),\boldsymbol{w}'')$ implies $\mu^{\mathrm{c}}(q',\pi'^{-1}(\boldsymbol{w}'^{I'}))=(q'',\pi''^{-1}(\boldsymbol{w}''^{I''}))$[4] and "conversely modulo I," $\mu^{\mathrm{c}}(q',\pi'^{-1}(\boldsymbol{w}'^{I'}))=(q'',\pi''^{-1}(\boldsymbol{w}''^{I''}))$ implies that for the unique $\boldsymbol{u}$ such that $\hat{\mu}^{\mathrm{c}}((q',I',\pi'),\boldsymbol{w}')=((q'',I'',\pi''),\boldsymbol{u}),\boldsymbol{u}^{I''}=\boldsymbol{w}''^{I''}$. Thus, $L(\hat{\boldsymbol{A}})=L(\boldsymbol{A})$. $\square$

8. Proof of II ⇒ I

Let Σ' be a finite subset of Σ and let $\equiv$ be a co-finitely insensitive and co-Σ'-invariant right congruence on Σ^* such that $\equiv^{\Sigma'}$ is of a finite index and L is a union of equivalence classes of $\equiv^{\Sigma'}$.

[4] As usual, π^{-1} denotes the inverse of permutation π. That is, $\pi\pi^{-1}=\mathrm{Id}$.

We start with some preliminary definitions and results.

By Proposition 3, $\equiv$ is co-r-insensitive for some non-negative integer r. With each $\boldsymbol{\sigma}\in\Sigma^*$ we associate an assignment $\boldsymbol{w}_\sigma = w_1 w_2 \dots w_{r+1} \in (\Sigma\cup\{\#\})^{r+1}$ as described below. We put $\boldsymbol{w}_\varepsilon = \#^{r+1}$, and $\boldsymbol{w}_{\boldsymbol{\sigma}\sigma} = w'_1 w'_2 \dots w'_{r+1}$ is obtained from $\boldsymbol{w}_{\boldsymbol{\sigma}} = w_1 w_2 \dots w_{r+1}$ in the following manner:

$$w'_{\min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}}))} = \begin{cases} \sigma & \text{if } \sigma \in \Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv} \backslash (\Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \Sigma'), \\ \# & \text{otherwise} \end{cases}$$

and for $i = 1, 2, \dots, r+1$, $i \neq \min(\#(\boldsymbol{w}_\sigma))$,

$$w'_i = \begin{cases} w_i & \text{if } w_i \in \Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv} \backslash \Sigma', \\ \# & \text{otherwise.} \end{cases}$$

Since $\equiv$ is co-r-insensitive, each of $\Sigma^{[\boldsymbol{\sigma}]_\equiv}$ and $\Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv}$ contains at most r elements. It follows by a straightforward induction on the length of $\boldsymbol{\sigma}$ that $\#(\boldsymbol{w}_{\boldsymbol{\sigma}}) \neq \emptyset$. Therefore, $\min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}}))$ and, consequently, $\boldsymbol{w}_{\boldsymbol{\sigma}\sigma}$ is well-defined. Also, by Proposition 4, $\Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv} \subseteq \Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \{\sigma\} \cup \Sigma'$. Thus, $\boldsymbol{w}_{\boldsymbol{\sigma}}$ is, indeed, a well-defined assignment.

Remark 4. It immediately follows from the definition above and Example 14 that $[\boldsymbol{w}_{\boldsymbol{\sigma}}] = \Sigma^{[\boldsymbol{\sigma}]_\equiv} \backslash \Sigma'$.

Roughly speaking, ("modulo Σ'"), $\boldsymbol{w}_{\boldsymbol{\sigma}}$ is the register contents of our automaton after reading input $\boldsymbol{\sigma}$.

Example 23. Let $\boldsymbol{\sigma}\in\Sigma^*$ and let L be as in Example 4. By Example 18, with respect to $\equiv_L$,

$$\boldsymbol{w}_{\boldsymbol{\sigma}} = \begin{cases} \sigma\# & \text{if } \boldsymbol{\sigma} = \sigma \in \Sigma, \\ \#\# & \text{otherwise.} \end{cases}$$

Proposition 5. *Let $F : [\boldsymbol{\sigma}] \to \Sigma$ be a Σ'-preserving function. Then $F(\boldsymbol{w}_{\boldsymbol{\sigma}}) = \boldsymbol{w}_{F(\boldsymbol{\sigma})}$.* [5]

Proof. The proof is by induction on the length of $\boldsymbol{\sigma}$. The basis immediately follows from the definition of $\boldsymbol{w}_\varepsilon$ and for the induction step let $\boldsymbol{\sigma}\in\Sigma^*$, $\sigma\in\Sigma$, and assume that $F(\boldsymbol{w}_{\boldsymbol{\sigma}}) = \boldsymbol{w}_{F(\boldsymbol{\sigma})}$. Let $\boldsymbol{w}_{\boldsymbol{\sigma}} = w_1 w_2 \dots w_{r+1}$ and $\boldsymbol{w}_{\boldsymbol{\sigma}\sigma} = w'_1 w'_2 \dots w'_{r+1}$. Then, $F(\boldsymbol{w}_{\boldsymbol{\sigma}}) = F(w_1)F(w_2)\dots F(w_{r+1})$, implying

$$\min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}})) = \min(\#(\boldsymbol{w}_{F(\boldsymbol{\sigma})})) \tag{6}$$

and $F(\boldsymbol{w}_{\boldsymbol{\sigma}\sigma}) = F(w'_1)F(w'_2)\dots F(w'_{r+1})$. By definition,

$$w'_{\min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}}))} = \begin{cases} \sigma & \text{if } \sigma \in \Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv} \backslash (\Sigma^{[\boldsymbol{\sigma}]_\equiv} \cup \Sigma'), \\ \# & \text{otherwise} \end{cases}$$

[5] Here and in the sequel we extend F onto a function from $[\boldsymbol{\sigma}] \cup \{\#\}$ into $\Sigma \cup \{\#\}$, also denoted F, by putting $F(\#) = \#$.

and for $i = 1, 2, \ldots, n + r,\ i \neq \min(\#(\boldsymbol{w_\sigma}))$,

$$w'_i = \begin{cases} w_i & \text{if } w_i \in \Sigma^{[\boldsymbol{\sigma}\sigma]_\equiv} \backslash \Sigma', \\ \# & \text{otherwise.} \end{cases}$$

Also

$$w'_{\min(\#(\boldsymbol{w}_{F(\boldsymbol{\sigma})}))} = \begin{cases} F(\sigma) & \text{if } F(\sigma) \in \Sigma^{[F(\boldsymbol{\sigma})F(\sigma)]_\equiv} \backslash (\Sigma^{[F(\boldsymbol{\sigma})]_\equiv} \cup \Sigma'), \\ \# & \text{otherwise} \end{cases}$$

and for $i = 1, 2, \ldots, n + r,\ i \neq \min(\#(\boldsymbol{w}_{F(\boldsymbol{\sigma})}))$,

$$w'_i = \begin{cases} F(w_i) & \text{if } F(w_i) \in \Sigma^{[F(\boldsymbol{\sigma})F(\sigma)]_\equiv} \backslash \Sigma', \\ \# & \text{otherwise.} \end{cases}$$

Since, by the induction hypothesis, $F(\boldsymbol{w_\sigma}) = \boldsymbol{w}_{F(\boldsymbol{\sigma})}$, the result follows from (6) and the above definitions of $\boldsymbol{w}_{\boldsymbol{\sigma}\sigma}$ and $\boldsymbol{w}_{F(\boldsymbol{\sigma}\sigma)}$. $\square$

For the construction of a DRFMA $\boldsymbol{A}$ that accepts L we shall need an equivalence relation $\sim$ on Σ^* that is defined by

$$\boldsymbol{\sigma}' \sim \boldsymbol{\sigma}'' \quad \text{if and only if } \boldsymbol{\sigma}' \equiv \boldsymbol{\sigma}'' \text{ and } \boldsymbol{w}_{\boldsymbol{\sigma}'} = \boldsymbol{w}_{\boldsymbol{\sigma}''}.$$

Example 24. Let L be as in Example 4. Then, by Example 23, $\sim_L$ coincides with $\equiv_L$, where $\sim_L$ is obtained from $\equiv_L$ like $\sim$ is obtained from $\equiv$.

Like in the classical case of a finite alphabet, states of $\boldsymbol{A}$ are equivalence classes of $\sim^{\Sigma'}$ and configurations of $\boldsymbol{A}$ are in one-to-one correspondence with equivalence classes of $\sim$. In particular, as we have already mentioned above, the register contents after reading a word $\boldsymbol{\sigma}$ is ("modulo Σ'") $\boldsymbol{w_\sigma}$. Actually, this is the reason for which we refined $\equiv$ to $\sim$. We did so to avoid the situation in which an equivalence class of $\equiv$ (an intended configuration) might correspond to a number of different register contents.

Proposition 6. *$\sim$ is a right congruence.*

Proof. Let $\boldsymbol{\sigma}' \sim \boldsymbol{\sigma}''$ and let $\sigma \in \Sigma$. It suffices to show that $\boldsymbol{\sigma}'\sigma \sim \boldsymbol{\sigma}''\sigma$. Since $\equiv$ is a right congruence, $\boldsymbol{\sigma}'\sigma \equiv \boldsymbol{\sigma}''\sigma$, and $\boldsymbol{w}_{\boldsymbol{\sigma}'\sigma} = \boldsymbol{w}_{\boldsymbol{\sigma}''\sigma}$ follows from $\Sigma^{[\boldsymbol{\sigma}']} = \Sigma^{[\boldsymbol{\sigma}'']}$, $\Sigma^{[\boldsymbol{\sigma}'\sigma]} = \Sigma^{[\boldsymbol{\sigma}''\sigma]}$, $\boldsymbol{w}_{\boldsymbol{\sigma}'} = \boldsymbol{w}_{\boldsymbol{\sigma}''}$, and the definition of $\boldsymbol{w}_{\boldsymbol{\sigma}'\sigma}$ and $\boldsymbol{w}_{\boldsymbol{\sigma}''\sigma}$. $\square$

Proposition 7. *$\sim$ is co-Σ'-invariant.*

Proof. Let $\boldsymbol{\sigma}' \equiv \boldsymbol{\sigma}''$ and $\boldsymbol{w}_{\boldsymbol{\sigma}'} = \boldsymbol{w}_{\boldsymbol{\sigma}''}$ and let $F : [\boldsymbol{\sigma}'] \cup [\boldsymbol{\sigma}''] \to \Sigma$ be a Σ'-preserving function. Since $\equiv$ is co-Σ'-invariant, $F(\boldsymbol{\sigma}') \equiv F(\boldsymbol{\sigma}'')$ and $\boldsymbol{w}_{F(\boldsymbol{\sigma}')} = \boldsymbol{w}_{F(\boldsymbol{\sigma}'')}$ follows from $\boldsymbol{w}_{\boldsymbol{\sigma}'} = \boldsymbol{w}_{\boldsymbol{\sigma}''}$ and Proposition 5. $\square$

Proposition 8. *$\sim^{\Sigma'}$ is of a finite index and L is the union of a number of equivalence classes of $\sim^{\Sigma'}$.*

Proof. Let $\Sigma^* = \bigcup_{i=1}^{n} [\boldsymbol{\sigma}_i]_{\equiv^{\Sigma'}}$. That is, for each $\boldsymbol{\sigma} \in \Sigma^*$ there is an $i = 1,2,\ldots,n$ and a Σ'-preserving function $F : [\boldsymbol{\sigma}] \to \Sigma$ such that $F(\boldsymbol{\sigma}) \in [\boldsymbol{\sigma}_i]_{\equiv}$.

By Remark 4, $[\boldsymbol{w}_{\boldsymbol{\sigma}}] = \Sigma^{[\boldsymbol{\sigma}]_{\equiv}} \backslash \Sigma'$, implying that each equivalence class of $\equiv$ is the union of at most $(r+1)^{r+1}$ equivalence classes of $\sim$. In particular, $[\boldsymbol{\sigma}_i]_{\equiv} = \bigcup_{j=1}^{n_i} [\boldsymbol{\sigma}_{i,j}]_{\sim}$ for some non-negative integer n_i and some $\boldsymbol{\sigma}_{i,j} \in \Sigma^*$, $j = 1,2,\ldots,n_i$. Therefore, $[\boldsymbol{\sigma}_i]_{\equiv^{\Sigma'}} = \bigcup_{j=1}^{n_i} [\boldsymbol{\sigma}_{i,j}]_{\sim^{\Sigma'}}$, $i = 1,2,\ldots,n$, implying $\Sigma^* = \bigcup_{i=1}^{n} \bigcup_{j=1}^{n_i} [\boldsymbol{\sigma}_{i,j}]_{\sim^{\Sigma'}}$.

Finally, since L is the union of a number of equivalence classes of $\equiv^{\Sigma'}$, L is the union of a number of equivalence classes of $\sim^{\Sigma'}$, as well. □

Now, let $\Sigma' = \{\sigma_1, \sigma_2, \ldots, \sigma_\ell\}$. Consider an $(r+1+\ell)$-register DRFMA $\boldsymbol{A} = \langle Q, q_0, \boldsymbol{w}_0, \mu, \mathscr{F}\rangle$, where

- Q is the set of all equivalence classes of $\sim^{\Sigma'}$,
- $q_0 = [\varepsilon]_{\sim^{\Sigma'}}$,
- $\boldsymbol{w}_0 = \#^{r+1}\sigma_1\sigma_2\ldots\sigma_\ell$ $(= \boldsymbol{w}_\varepsilon\sigma_1\sigma_2\ldots\sigma_\ell)$,
- $\mathscr{F} = \{[\boldsymbol{\sigma}]_{\sim^{\Sigma'}} : \boldsymbol{\sigma} \in L\}$,

and for $\boldsymbol{\sigma} \in \Sigma^*$ and $i = 1,2,\ldots,r+1+\ell$, $\mu([\boldsymbol{\sigma}]_{\sim^{\Sigma'}}, i) = (\#(\boldsymbol{w}_{\boldsymbol{\sigma}\sigma}), [\boldsymbol{\sigma}\sigma]_{\sim^{\Sigma'}})$, where σ is defined as follows.

Let $\boldsymbol{w}_{\boldsymbol{\sigma}} = w_1, w_2, \ldots, w_{r+1}$. Then

$$\sigma = \begin{cases} \sigma_{i-r-1} & \text{if } i > r+1, \\ w_i & \text{if } i \leqslant r+1 \text{ and } w_i \neq \#, \\ \text{any element in } \Sigma\backslash([w_{\boldsymbol{\sigma}}] \cup \Sigma') & \text{otherwise.} \end{cases} \tag{7}$$

Remark 5. Note that in the case of $i = \min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}}))$ in (7), $[\boldsymbol{\sigma}\sigma]_{\sim^{\Sigma'}}$ and $\#(\boldsymbol{w}_{\boldsymbol{\sigma}\sigma})$ do not depend on the choice of $\sigma \in \Sigma\backslash([w_{\boldsymbol{\sigma}}] \cup \Sigma')$. Indeed, let $\sigma', \sigma'' \in \Sigma\backslash([w_{\boldsymbol{\sigma}}] \cup \Sigma')$ and let $F : [\boldsymbol{\sigma}\sigma'] \to \Sigma$ be defined by

$$F(\sigma) = \begin{cases} \sigma & \text{if } \sigma \neq \sigma', \\ \sigma'' & \text{if } \sigma = \sigma'. \end{cases}$$

Then $\boldsymbol{\sigma}\sigma'' = F(\boldsymbol{\sigma}\sigma')$. By definition, F is Σ'-preserving. Therefore, $\boldsymbol{\sigma}\sigma' \sim^{\Sigma'} \boldsymbol{\sigma}\sigma''$, implying $[\boldsymbol{\sigma}\sigma']_{\sim^{\Sigma'}} = [\boldsymbol{\sigma}\sigma'']_{\sim^{\Sigma'}}$. Finally, by Proposition 5, $\boldsymbol{w}_{\boldsymbol{\sigma}\sigma''} = F(\boldsymbol{w}_{\boldsymbol{\sigma}\sigma'})$, whence $\#(\boldsymbol{w}_{\boldsymbol{\sigma}\sigma'}) = \#(\boldsymbol{w}_{\boldsymbol{\sigma}\sigma''})$.

We have to prove that μ is well-defined. That is, the value of μ does not depend on a particular representative of an equivalence class. Let $\boldsymbol{\sigma}' \sim^{\Sigma'} \boldsymbol{\sigma}''$, $\mu([\boldsymbol{\sigma}']_{\sim^{\Sigma'}}, i) = (\boldsymbol{w}_{\boldsymbol{\sigma}'\sigma'}, [\boldsymbol{\sigma}'\sigma']_{\sim^{\Sigma'}})$, and $\mu([\boldsymbol{\sigma}'']_{\sim^{\Sigma'}}, i) = (\boldsymbol{w}_{\boldsymbol{\sigma}''}\sigma'', [\boldsymbol{\sigma}''\sigma'']_{\sim^{\Sigma'}})$. We shall show that $\boldsymbol{w}_{\boldsymbol{\sigma}'\sigma'} = \boldsymbol{w}_{\boldsymbol{\sigma}''\sigma''}$ and $\boldsymbol{\sigma}'\sigma' \sim^{\Sigma'} \boldsymbol{\sigma}''\sigma''$.

Let $\boldsymbol{w}_{\boldsymbol{\sigma}'} = w'_1 w'_2 \ldots w'_{r+1}$, $\boldsymbol{w}_{\boldsymbol{\sigma}''} = w''_1 w''_2 \ldots w''_{r+1}$, Since $\boldsymbol{\sigma}' \sim^{\Sigma'} \boldsymbol{\sigma}''$, there exists a Σ'-preserving function $F : [\boldsymbol{\sigma}'] \to \Sigma$ such that

$$F(\boldsymbol{\sigma}') \sim \boldsymbol{\sigma}''. \tag{8}$$

We shall distinguish between the cases of $i > r+1$, $i \leqslant r+1$ and $i \neq \min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}'}))$, and $i = \min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}'}))$.

Assume $i > r+1$. Then $\sigma' = \sigma'' = \sigma_{i-r-1}$. By (8), $F(\boldsymbol{\sigma}') = \boldsymbol{\sigma}''$, implying

$$F(\boldsymbol{\sigma}'\sigma') = F(\boldsymbol{\sigma}')\sigma'' \sim \boldsymbol{\sigma}''\sigma'', \tag{9}$$

because, by Proposition 6, $\sim$ is a right congruence.

Next,

$$\#(\boldsymbol{w}_{\boldsymbol{\sigma}'\sigma'}) = \#(\boldsymbol{w}_{F(\boldsymbol{\sigma}'\sigma')}) = \#(\boldsymbol{w}_{\boldsymbol{\sigma}''\sigma''}),$$

where the first equality is by Proposition 5 and the second equality follows from (9) and the definition of $\sim$.

The case of $i \neq \min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}'}))$, $i \leqslant r+1$ is treated similarly to the above. In this case $\sigma' = w_i'$ and $\sigma'' = w_i''$. Again, by (8), $F(\boldsymbol{\sigma}') = \boldsymbol{\sigma}''$, implying $F(\boldsymbol{\sigma}'\sigma') = F(\boldsymbol{\sigma}')\sigma'' \sim \boldsymbol{\sigma}''\sigma''$, because, by Proposition 6, $\sim$ is a right congruence.

We omit the proof of $\#(\boldsymbol{w}_{\boldsymbol{\sigma}'\sigma'}) = \#(\boldsymbol{w}_{\boldsymbol{\sigma}''\sigma''})$, because it is exactly as in the case of $i > r+1$.

Finally, let $i = \min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}'}))$. Since, by Proposition 5, $\boldsymbol{w}_{\boldsymbol{\sigma}''} = F(\boldsymbol{w}_{\boldsymbol{\sigma}'})$, $i = \min(\#(\boldsymbol{w}_{\boldsymbol{\sigma}''}))$ as well. Then $\sigma' \notin \Sigma' \cup [\boldsymbol{w}_{\boldsymbol{\sigma}'}]$ and $\sigma'' \notin \Sigma' \cup [\boldsymbol{w}_{\boldsymbol{\sigma}''}]$.

We extend F onto $[\boldsymbol{\sigma}'\sigma']$ by putting $F(\sigma') = \sigma''$. Then, since by Proposition 6, $\sim$ is a right congruence, $F(\boldsymbol{\sigma}'\sigma') = F(\boldsymbol{\sigma}')\sigma'' \sim \boldsymbol{\sigma}''\sigma''$. That is, $\boldsymbol{\sigma}'\sigma' \sim^{\Sigma'} \boldsymbol{\sigma}''\sigma''$.

Again, we omit the proof of $\#(\boldsymbol{w}_{\boldsymbol{\sigma}'\sigma'}) = \#(\boldsymbol{w}_{\boldsymbol{\sigma}''\sigma''})$, because it is exactly as in the case of $i > r+1$.

Now a straightforward induction on the length of $\boldsymbol{\sigma} \in \Sigma^*$ shows that $\mu^c(\boldsymbol{\sigma}) = ([\boldsymbol{\sigma}]_{\sim_{\Sigma'}}, \boldsymbol{w}_{\boldsymbol{\sigma}})$, which, in turn, implies the desired equality $L = L(\boldsymbol{A})$.

We conclude this section with an example of constructing a DRFMA from an equivalence relation that satisfies condition II of Theorem 1.

Example 25. Let L be as in Example 4. By Examples 12 and 19, $\equiv_L$ is co-$\emptyset$-invariant and co-1-insensitive. In addition, it follows from Examples 4 and 24 that $\equiv_L^{\emptyset}$ (and, consequently, $\sim_L^{\emptyset}$) has four equivalence classes: $\{\varepsilon\}$, Σ, L, and $\Sigma^* \setminus (L \cup \Sigma \cup \{\varepsilon\})$, denoted by q, p, f, and s, respectively. These are the states of a DRFMA accepting L that we constructed in the proof of II $\Rightarrow$ I. In particular, q and f are the initial and the final states, respectively. Finally, by Example 23, the transition function of the above DRFMA is given by

State/register	1	2
q	$(\{2\}, p)$	$(\{2\}, p)$
p	$(\{1,2\}, f)$	$(\{1,2\}, s)$
f	$(\{1,2\}, s)$	$(\{1,2\}, s)$
s	$(\{1,2\}, s)$	$(\{1,2\}, s)$

,

cf. Example 22.

9. Discussion of future research

We conclude the paper with some problems which, on one hand, are of interest in their own right, and, on the other hand, might give a better insight into deterministic quasi-regular languages.

- The most natural question to ask is the minimality of deterministic finite-memory automata. Is the DFMA constructed from $\equiv_L$ the minimal one and in which sense? Here an additional problem is that it is not clear how the minimality should be defined: with respect to the number of states or with respect to the number of registers (or both)?
- The following problem is tightly related to the first one. Does $\sim$ obtained from $\equiv_L$ coincide with $\equiv_L$ if L is quasi-regular (cf. Example 24)? If so, the FMA constructed from $\equiv_L$ would be minimal with respect to the number of states.
- Does the co-Σ'-invariance of $\equiv_L$ imply the co-Σ'-invariance of L? An affirmative answer would strengthen Theorem 1 by replacing the co-Σ'-invariance of L with the co-Σ'-invariance of $\equiv_L$ in clause II and the negative answer would show that clause II cannot be relaxed.
- Is it decidable whether a quasi-regular language is deterministic?
- Finally we repeat the problem of the relationship between deterministic and non-deterministic quasi-regular languages we stated in [10]: does each quasi-regular language belong to the closure of deterministic quasi-regular languages under union, intersection, concatenation, and Kleene star?

References

[1] J.-M. Autebert, J. Beauquir, L. Boasson, Langages des alphabets infinis, Discrete Appl. Math. 2 (1980) 1–20.
[2] J.-M. Autebert, J. Beauquir, L. Boasson, Formes de langages et de grammaries, Acta Inform. 17 (1982) 193–213.
[3] M. Bielecki, J. Hidders, J. Paredaens, J. Tyszkiewicz, J. Van den Bussch, Navigating with a browser, in: P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, R. Conejo (Eds.), Proc. 29th Internat. Colloq. on Automata, Languages and Programming—ICALP 2002, Lecture Notes in Computer Science, Vol. 2380, Springer, Berlin, 2002, pp. 764–775.
[4] B. Bolling, M. Leucker, t. Noll, Generalized regular MSA languages, Technical Report, Department of Computer Science, Aachen University of Technology, 2002.
[5] E.Y.C. Cheng, M. Kaminski, Context-free languages over infinite alphabets, Acta Inform. 35 (1998) 245–267.
[6] A. Ginzburg, Algebraic Theory of Automata, Academic Press, New York, NY, 1968.
[7] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 1979.
[8] J. Itd, Automates a pile sur des alphabets infinis, in: Proc. Symp. of Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, Vol. 166, Springer, Berlin, 1984, pp. 260–273.
[9] M. Kaminski, N. Francez, Finite-memory automata, in: Proc. 31th Annu. IEEE Symp. on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitis, CA, 1990, pp. 683–688.
[10] M. Kaminski, N. Francez, Finite-memory automata, Theoret. Comput. Sci. A 138 (1994) 329–363.
[11] F. Neven, T. Schwentik, V. Vianu, Towards regular languages over infinite alphabets, in: J. Sgall, A. Pultr, P. Kolman (Eds.), Proc. 26th Internat. Symp. on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Vol. 2136, Springer, Berlin, 2001, pp. 560–572.

[12] F. Otto, Classes of regular and context-free languages over countably infinite alphabets, Discrete Appl. Math. 12 (1985) 41–56.
[13] M.O. Rabin, D. Scott, Finite automata and their decision problems, IBM J. Res. Develop. 3 (1959) 114–125.
[14] H. Sakamoto, D. Ikeda, Intractability of decision problems for finite-memory automata, Theoret. Comp. Sci. A 231 (2000) 297–308.
[15] Y. Shemesh, N. Francez, Finite-state unification automata and relational languages, Inform. and Comput. 114 (1994) 192–213.